Evolutionary image analysis and signal processing

Stefano Cagnoni

Intelligent Bio-Inspired System (IBIS) Laboratory
Department of Information Engineering
University of Parma, Italy

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2662-9/14/07.

Instructor

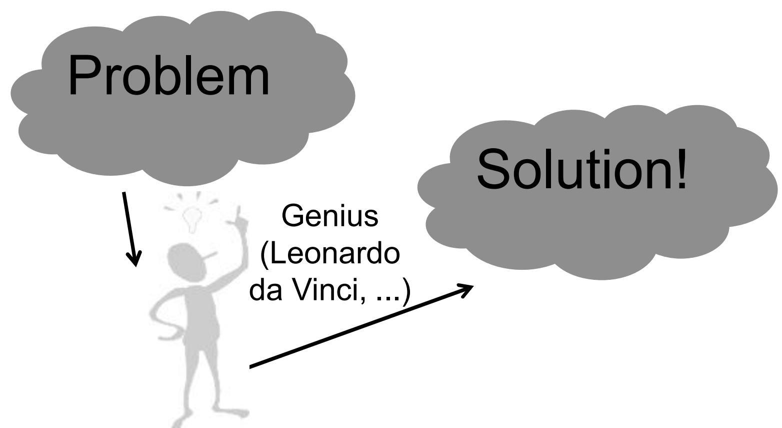
Stefano Cagnoni works at the University of Parma, where he has been Associate Professor since 2004.

Recent research grants include: co-management of a project funded by Italian Railway Network Society (RFI) aimed at developing an automatic inspection system for train pantographs, and a "Marie Curie Initial Training Network" grant, for a four-year research training project in Medical Imaging using Bio-Inspired and Soft Computing.
Editor-in-chief of the "Journal of Artificial Evolution and Applications" from 2007 to 2010. Since 1999, he has been chair of EvoIASP, an event dedicated to evolutionary computation for image analysis and signal processing, now a track of the EvoApplications conference. Since 2005, he has co-chaired MedGEC, workshop on medical applications of evolutionary computation at GECCO. Co-editor of special issues of journals dedicated to Evolutionary Computation for Image Analysis and Signal Processing. Member of the Editorial Board of the journals "Evolutionary Computation" and "Genetic Programming and Evolvable Machines".

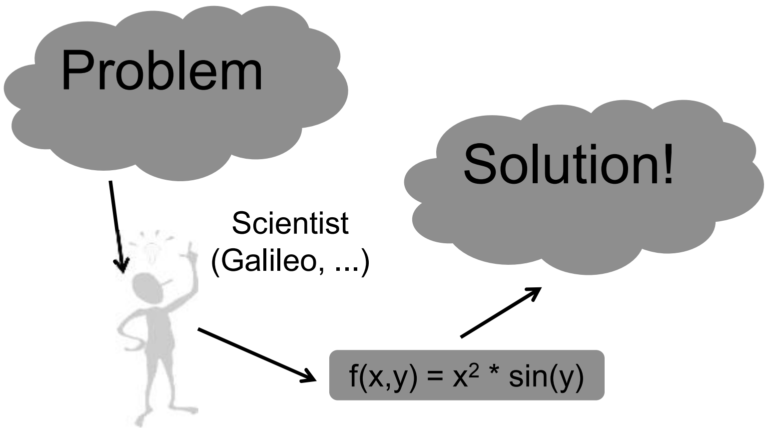
Overview

- Design and optimization
- Metaheuristics (MHs)
- MHs and image analysis: Evolutionary Computation and Swarm Intelligence
  - MHs as general optimization tool
    - Examples
  - GPU-based parallel implementations
  - MHs for model-based object detection and tracking
    - Examples

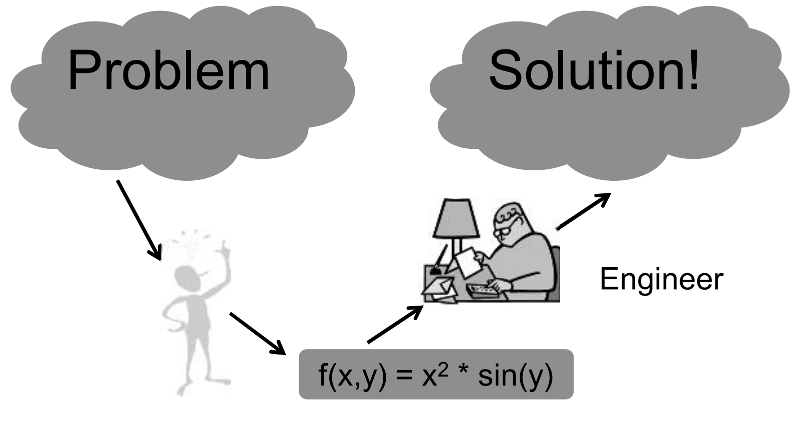
A short history of design - I



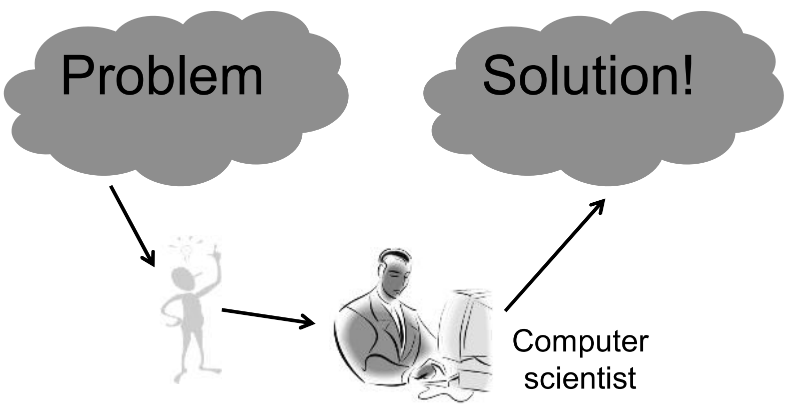
A short history of design - II



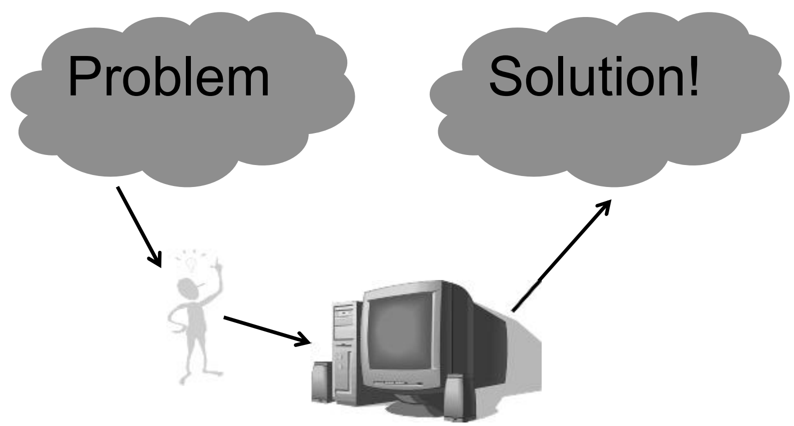
A short history of design - III



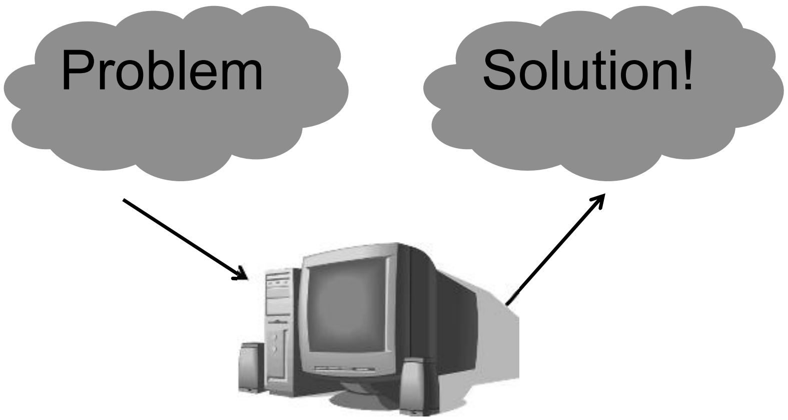
A short history of design - IV



A short history of design - V



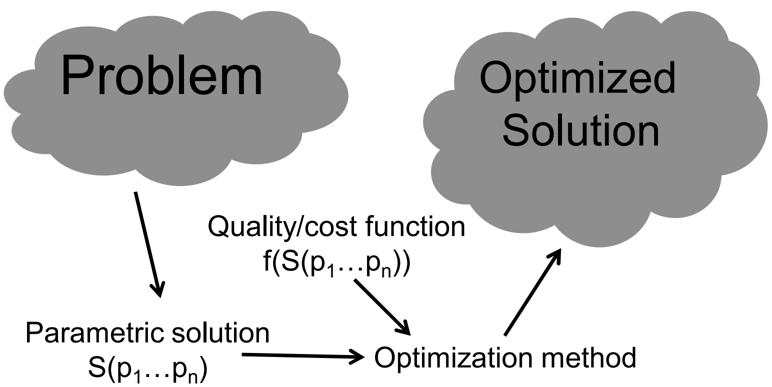
A short history of design – next step?



Design = Optimization

- Most real-world problems can be formulated as optimization problems.
- Designers know what they want to obtain (output), and can measure success quantitatively.
- The available degrees of freedom (inputs, independent parameters, etc.) are also known.
- ***The I/O mapping is only partially known.***
- Approximate models may be available.

Typical optimization scenario



Design by optimization

The quality function to be optimized typically:

- relies on an **ideal** model of a practical problem, of which a parametric solution is given
- is typically optimized based on the performance achieved on a set of **real** samples of the problem at hand

An optimization technique is finally used to find the best parameters for the solution (which maximize the quality function on the real data)

A shifted view upon design

We are currently in phase V.

Design often shifts from

- ***defining* exact** solutions, **justified by** an underlying **theory**

to

- ***searching*** solutions which ***work well,*** by:
  - defining a quality criterion that measures the effectiveness (cost) of possible solutions
  - choosing a method that maximizes/minimizes it.

A shifted view upon design

Next step, or final goal for computer research would be to switch the engineer's attention from

- tuning the parameters of a *specific* solution to a problem using knowledge about the problem

to

- tuning the parameters of a *general* optimization method using knowledge about the method or, even better, letting the computer adapt them itself

When would this be necessary ?

- No direct solution is available
- Problem specifications provided only qualitatively or through examples
- Behaviors or phenomena described or measured with little precision (e.g., noisy signals)
- Little a priori knowledge (none ?) on the problem
- Integration of heterogeneous modules to which any of the previous conditions applies

When would this be necessary ?

In general:

- the less knowledge is available about the problem (i.e., the more one deals with **black-box optimization** problems)
- the more general and stochastic the algorithm used to solve them

Metaheuristics (MHs)

Optimization methods which iteratively improve a (set of) candidate solution(s) with respect to a given measure of quality.

MHs make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

No optimal solution is guaranteed to be found.

Many, such as Evolutionary Computation and Swarm Intelligence techniques, are inspired by nature.

MH-based applications taxonomy

According to the abstraction level of the application

- Low-level approaches (e.g., design of filters)
- Mid-level approaches (design of classifiers)
- High-level approaches (model-based object detection)

The solution is usually designed first.
Then a MH is used to optimize/generate a part of it.

MH-based applications taxonomy

According to the MH used

- Optimization of parameters of *pre-defined problem-specific objective functions*

  Related with a well-defined task or for a whole system.
- Generation of solutions from scratch, based on *pre-defined operands or building blocks.*

  No predefined structure for the solution. Problem-specific objective functions.

Again, MHs are 'external tools'

MH-based applications taxonomy

According to the role of MHs

- MH as external optimization tools
- Interactive generation of solutions
- ***Generation of emergent collective solutions***

Achievement of higher-level tasks by complex behaviors emerging from collective use of trivial, local, hard-wired ones: complete solutions embedding MHs

MHs are no more seen as parameter optimizers or 'external' tools but as (part of) THE solution*!*

APPLICATIONS TO SIGNAL/IMAGE PROCESSING AND PATTERN RECOGNITION

- Optimization of filter/detector AND algorithm parameters for event detection in 1D signals and for 3D image segmentation
- SI-based object detection and tracking
- Model-based object detection and segmentation

SIGNAL PROCESSING

- Signal Enhancement (filtering) and Event Detection (thresholding)

EVOLUTIONARY DESIGN OF QRS DETECTORS

Given:

- Filter/detector layout
- Training set
- Fitness function

Optimize (using a GA):

- Filter coefficients
- Detector threshold
- Other parameters regulating the adaptive behavior of the detector

FILTER LAYOUT

- Linear:

$$y_i = a_0 + \sum_{k=1}^{10} a_k x_{i-k}$$

- Linear with selected samples:

$$y_i = a_0 + \sum_{k=1}^{5} a_k x_{i-d_k}$$

- Quadratic with selected samples:

$$y_i = \underbrace{\sum_{k_1=0}^{2}\sum_{k_2=0}^{2}\sum_{k_3=0}^{2}}_{\sum k_j \le 2} a_{k_1k_2k_3} x_{i-d_1}^{k_1} x_{i-d_2}^{k_2} x_{i-d_3}^{k_3} \qquad d_i \in (1,10),\ \ d_{i+1} > d_i$$

OTHER PARAMETERS

Y_i = adaptive threshold, such that

$$Y_0 = Y_{start}$$
$$Y_i = g((1-\alpha)Y_{i-1} - \beta z_{i-1}(Y_{i-1} - \gamma y_{i-1}))$$

where $g(x) = \max\{Y_{min}, \min[Y_{max}, x]\}$

α = decay rate

β = speed with which Y_i moves towards γy_{i-1}.
γ = percentage of last peak towards which the threshold decays.

EXPERIMENTAL SETUP

TRAINING SET

10 10-second tracts of the ECG from each of the 48 30-minute records of the MIT-BIH Arrhythmia Database (5981 beats out of about 110,000).

FITNESS FUNCTION

$f = f_{max} - (FP^2 + FN^2)$, f_{max} such that $f>0$

FP = False Positives, FN = False Negatives

RESULTS

- 99.5% average sensitivity (100% on most “normal” recordings)
- Much faster detection with respect to published algorithms yielding comparable results or better results with comparable computational effort

APPLICATIONS TO SIGNAL/IMAGE PROCESSING AND PATTERN RECOGNITION

- Optimization of filter/detector AND algorithm parameters for event detection in 1D signals and for 3D image segmentation
- SI-based object detection and tracking
- Model-based object detection and segmentation

Tracking of anatomical structures

Tomographic images

Extraction of the same structure in consecutive sections to recover the whole 3D structure

Exploitation of the correlation between consecutive sections

APPLICATIONS TO SIGNAL/IMAGE PROCESSING AND PATTERN RECOGNITION

Optimization of filter/detector AND algorithm parameters for 3D image segmentation

- Adaptive filtering: filter coefficients are learnt based on the ‘observation’ of manual segmentations
- Relevant points are detected
- The structure’s shape is recovered using a deformable model
- The process is iterated over the whole image stack

RATIONALE

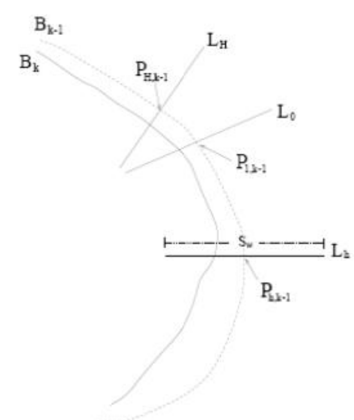
Pre-defined parameter sets hardly work in the presence of high variability, as in biology/anatomy

- GA optimization of a specific structure segmentation algorithm
- interactive specification of a few training contours, followed by
- extraction of the contours of the structure of interest from the whole data set

CONTOUR EXTRACTION

The problem can be reformulated as ‘multiple 1D-edge detection and tracking’.

An extension of the 1D detector



SEGMENTATION

- Definition of a starting contour
- Iterate:
  - Application of the GA-designed filter to the next contour (extraction of matching edge points)
  - Elastic contour model-based interpolation (also optimized by the GA) of the edge points extracted by the filter

TRAINING SET

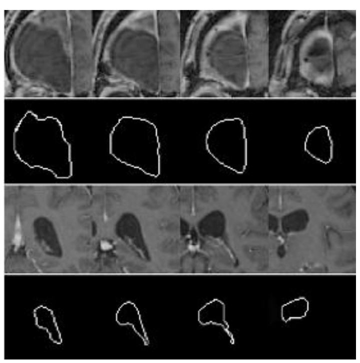
One slice following the one which is used to seed the iterative segmentation process

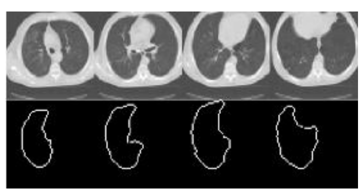
FITNESS FUNCTION

$$F(\{y\}) = K - \sqrt{\sum_{k=1}^{H} d_{y_k}{}^2}$$

d_{yk} = distance, along scan line L_h, between the actual edge point and the one detected, K = constant

RESULTS





APPLICATIONS TO SIGNAL/IMAGE PROCESSING AND PATTERN RECOGNITION

- Optimization of filter/detector AND algorithm parameters for event detection in 1D signals and for 3D image segmentation
- SI-based object detection and tracking
- Model-based object detection and segmentation

SWARM INTELLIGENCE FOR IMAGE ANALYSIS

- Particle Swarm Optimization
- An application: license plate detection

Why PSO ?

- Easy to implement
- Very well parallelizable (very few dependencies)
- Fast convergence to a good basin of attraction (usually less effective at refining the search)
- Exploration driven by easy physical laws
- Intrinsic capacity of tracking solutions in a time-varying environment

PSO and Image Analysis

- **Multi-objective optimization:** the goal is to identify (segment out) one or **more** regions of interest (ROIs) in the search space (the image), not a single optimum.
  - k-mean clustering PSO (Passaro and Starita 2006): the swarm reorganizes itself in multiple sub-swarms: each sub-swarm may then 'address' a different target.
- The swarm must then **cover** the region as uniformly as possible

PSO-based License Plate Detection

- Plates feature high density of pixels with high horizontal gradient values due to the contrast between letters (black) and plate background (white)
- The horizontal gradient can be easily and efficiently approximated using differences

PSO-based License Plate Detection

Search stages :

- **Global search:** the most ‘promising’ areas within the whole image are grossly detected

- **Local search:** attention is focused on the ROIs defined in the previous stage to detect good plate candidates

PSO-based License Plate Detection

- A *repulsion* term is added to let the swarm spread all over the target:

$$v_P^*(t) = v_P(t) + repulsion_p$$

- Repulsion between two particles is computed, separately along each axis, as

$$repulsion(i, j) = REPULSION\_RANGE - |X_i - X_j|$$

REPULSION_RANGE being the maximum distance within which interaction between particles occurs.

- The global repulsion term for P is the average of all repulsion terms with particles interacting with it:

$$repulsion_p = \sum_j repulsion(p,j) / n$$

PSO-based License Plate Detection

- The goal is to find regions featuring high density of high-gradient pixels
- To prevent swarms from converging towards ISOLATED pixels, fitness has two terms:
  - *punctual fitness*, depending on visual features
  - *local fitness*, proportional to the number of neighbors

PSO-based License Plate Detection

- The swarm explores a gray-scale image

fitness(x,y) = punctual fitness(x,y) + local fitness(x,y)

if (|r(x,y)-g(x,y)|< K_0 && |r(x,y)-b(x,y)|<K_0 && |g(x,y)-b(x,y)| <K_0)

punctual_fitness = *max(right_gradient,left_gradient);*

else **punctual_fitness** = *0;*

where right_gradient = |grayscale(x,y)-grayscale(x+1,y)|;

left_gradient = |grayscale(x,y)-grayscale(x-1,y)|;

local fitness = *K1 * neighbor_number*

PSO-based License Plate Detection

- The standard equation has been further modified, to increase stability of sub-swarms.
- If a particle with high punctual fitness lies within a region with high density of particles, then it has a probability, which is linearly dependent on such a density, of staying there:

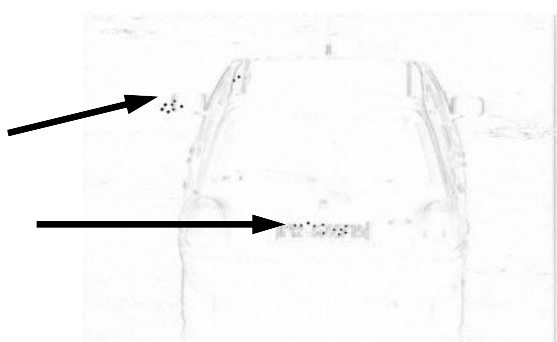
Prob { $X_{t-1}=X_t$ } = *number of particles in the neighborhood / total number of particles in the swarm*

PSO-based License Plate Detection

GLOBAL SEARCH RESULTS

PSO-based License Plate Detection

Two ‘promising’ areas are detected (sub-swarms with *Nparticles* > *T*): the larger one will be explored first



PSO-based License Plate Detection

The whole swarm is re-initialized near the selected area and a local search is performed, using the previous algorithm.

PSO-based License Plate Detection

A bounding box is computed enclosing all particles with high punctual fitness. If this box has a *w:h ratio = 5:1*, we can assert we found the plate.

PSO-based License Plate Detection

Otherwise the box is expanded, letting some agent move up and down (or left and right), in order to reach the given ratio. On failure, the next region is explored.

PSO-based License Plate Detection

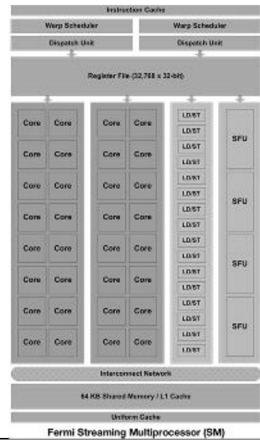
Final result

WHAT ABOUT GOING REAL-TIME (OR INCREASING TASK COMPLEXITY) ?

- GPU-based implementation of EC/SI algorithms

GPU

- Massively parallel architecture
  - Hundreds or thousands of simple cores
  - Simple instruction set
    - Synchronization primitives
- Deep memory hierarchy
  - Private, local, global, constant memory
  - Each one has a different role



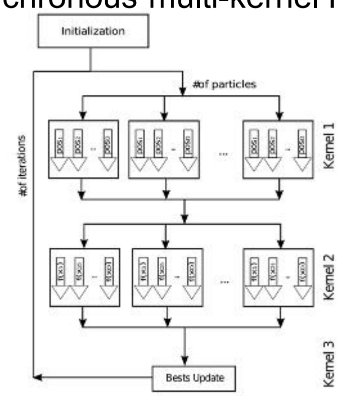
GPU-based PSO implementations

- Three-kernel synchronous (Information Sciences, 2011)
  - Any topology allowed
  - Any problem size
  - Large overhead (three memory swaps)
- Single-kernel asynchronous (GECCO 2011)
  - Ring topology, radius = 1
  - Limited number of particles
  - Fastest possible (no swaps)

GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

Single-kernel vs. Multi-kernel

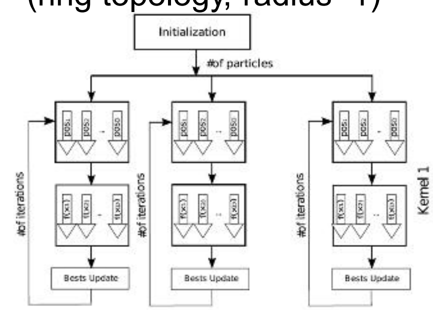
- Synchronous multi-kernel PSO



GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

Single-kernel vs. Multi-kernel

Asynchronous single-kernel PSO
(ring topology, radius=1)



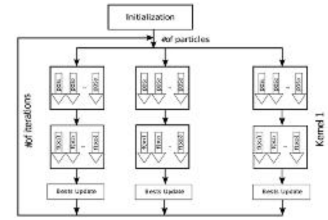
GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

Single-kernel vs. Multi-kernel

- Single-kernel (all computations in local memory)
  - \+ No (limited) need for synchronization
    No data exchange between GPU and CPU
  - \- Limited local resources
    Small maximum number of particles in a swarm
- Multi-kernel (need for 3 data swaps)
  - \+ Virtually no resource-related limitation
    Any swarm size possible (up to several hundreds)
  - \- Large memory overhead due to the need for synchronization after each kernel is run

More general implementation

- Single kernel
- Synchronization at the end of each cycle
  - One can schedule as many threads as necessary
- Suitable for both CPUs & GPUs
- Virtually no limits to the number of particles
- Smaller memory overhead wrt the multi-kernel version



APPLICATIONS TO SIGNAL/IMAGE PROCESSING AND PATTERN RECOGNITION

- Optimization of filter/detector AND algorithm parameters for event detection in 1D signals (background)
- Optimization of filter/detector AND algorithm parameters for 3D image segmentation
- Model-based object detection and segmentation

Model-based detection

- Traditional Image Analysis Approach: from image to object
  - The image is exhaustively and mechanically scanned using a moving window until something interesting is found

Model-based detection

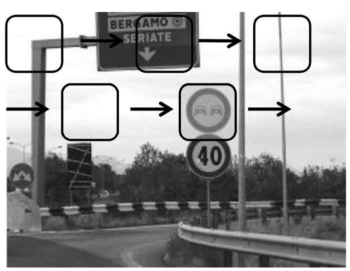
- Model-based approach: from object (model) to image:
  - A parametric model describing the possible variations of the object is defined along with its projection on the image plane
  - An optimization method finds the set of parameters which maximizes a similarity measure with the actual image content

Road Sign Recognition

- Usually divided into:
  - Sign Detection
    (color-based or shape-based approaches)
  - Sign Classification
    (usually based on Artificial Neural Networks)

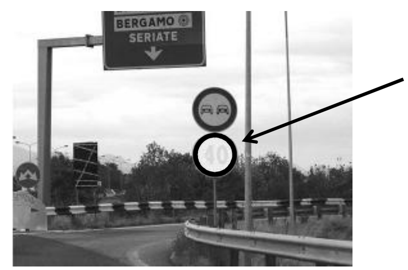
We propose an inverse approach to sign detection
  - 3D model based estimation

Classical Road Sign Detection



- Exhaustive window scan, with different box sizes
- Classifier-based detection

Our Approach - 1



- Detecting signs and estimating poses at the same time
- Hypothesizing a sign pose with respect to a calibrated camera, a sign model can be projected onto the image and matched to the actual image content

Our Approach - 2

- Making assumptions on the sign pose can help discard visible signs which can be ignored by the driver
- For example, these signs must be considered only by those who take the right exit

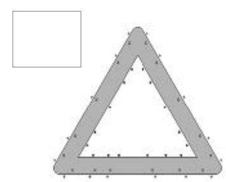
How to find a good estimation quickly?

- Particle/Kalman Filtering is usually too slow to achieve real-time performances...
- We propose an approach based on a bio-inspired metaheuristic called Particle Swarm Optimization (PSO)
- PSO searches for function extrema mimicking the behaviour of a flock of birds in search of food

PSO for Road Sign Detection

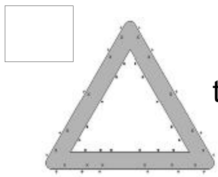
- **Search domain** = space of the roto-translation matrices
- **Fitness function** proportional to the presence of local features which characterize the model of our target in the image region onto which the model is projected

PSO for Road Sign Detection

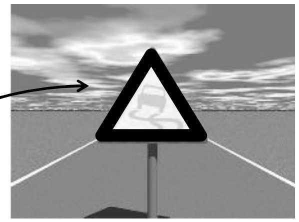


- 6 parameters: sign roto-translation wrt the camera
- The parameters can be reduced to 4 (x, y, z and yaw)
- Image is accessed (sampled) only locally during fitness evaluation: no need for an exhaustive scan

PSO for Road Sign Detection

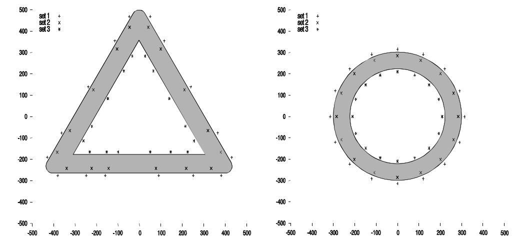


•Roto-translation and projection



- 6 parameters: sign roto-translation wrt the camera
- The parameters can be reduced to 4 (x, y, z and yaw)
- Image is accessed (sampled) only locally during fitness evaluation: no need for an exhaustive scan

PSO for Road Sign Detection



- Model: Three different sets of 3D points, describing the sign shape and regions: border, inside, outside
- The likelihood of detection is computed by matching color histograms

PSO for RSD - Fitness Function

$$f = 1 - \frac{k_0\ (1 - S_{1,2}) + k_1\ (1 - S_{2,3}) + k_2\ S_{1,ref}}{k_0 + k_1 + k_2}$$

$$S_{x,y} = \frac{\rho(\boldsymbol{H}_x^R, \boldsymbol{H}_y^R) + \rho(\boldsymbol{H}_x^G, \boldsymbol{H}_y^G) + \rho(\boldsymbol{H}_x^B, \boldsymbol{H}_y^B)}{3}$$

$$\rho(\boldsymbol{H}_1, \boldsymbol{H}_2) = \sum_{b=1}^{N_{bin}} \sqrt{H_1(b) H_2(b)}$$

•Bhattacharyya Similarity

- The fitness function is based on the Bhattacharyya Similarity between reference (model) histograms and the histograms of the image region onto which the model is projected

PSO for RSD - Fitness Function

$$f = 1 - \frac{\overset{1}{\boxed{k_0\ (1 - S_{1,2})}} + \overset{2}{\boxed{k_1\ (1 - S_{2,3})}} + \overset{3}{\boxed{k_2\ S_{1,ref}}}}{k_0 + k_1 + k_2}$$

To detect warning (triangular) and regulatory (round) signs we maximize:

1. difference between the red border and background
2. the difference between the white inside and the red border
3. similarity between red band and a reference red color

Road Sign Detection within CUDA

<initialize positions/velocities of all particles>
<perform a first evaluation of the fitness functions>
<set initial personal/global bests>

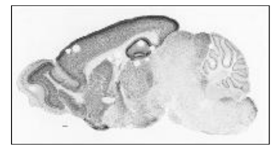
for(int i = 0; i < generationsNumber; i++){
<update the position of all particles>
<re−evaluate the fitness of all particles>
<update all personal/global bests>
}

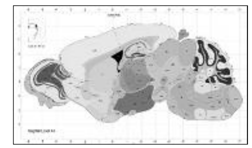
•Point-projection operations and histogram computation/comparison could also be parallelized

<retrieve global best information to be returned as final result>

Hippocampus Localization

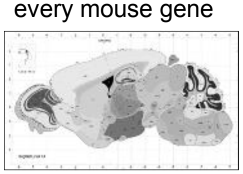
- Allen Mouse Brain Atlas
  - Public database of high-resolution brain images
    - Expression patterns of about 20,000 genes in the adult mouse brain
    - Spatial map of the expression patterns of almost every mouse gene

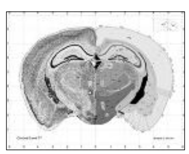

NISSL


REFERENCE

Hippocampus Localization

- Allen Mouse Brain Atlas
  - Public database of high resolution brain images
    - Expression patterns of about 20,000 genes in the adult mouse brain
    - Spatial map of the expression patterns of almost every mouse gene

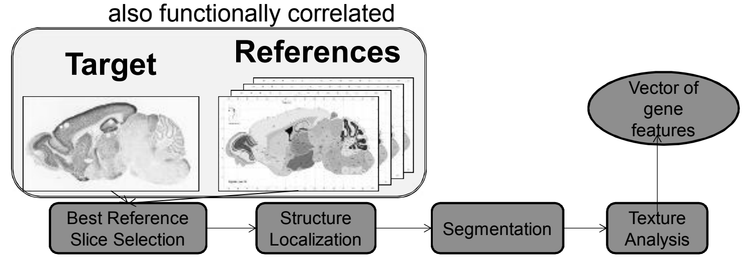

21 SAGITTAL


132 CORONAL

Hippocampus Localization

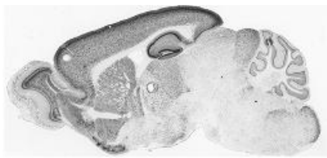
- Final Goal
  - Extract a feature vector for each gene which allows one to **cluster genes** into similar subsets

    Hypothesis: genes with similar expression patterns are also functionally correlated



Hippocampus Localization

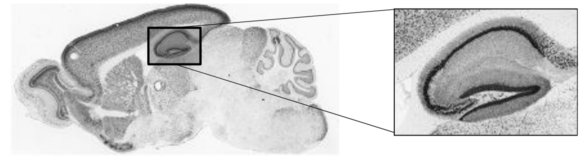
- Automatic localization of brain structures in tomographic images
  - Allen Brain Atlas (mouse brain histological images)



GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

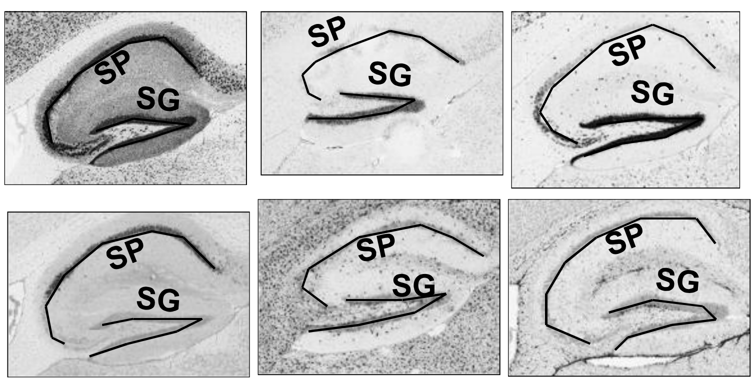
Hippocampus Localization

- Automatic localization of brain structures in tomographic images
  - Allen Brain Atlas (mouse brain histological images)
  - Hippocampus (important role in learning and memory)



GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

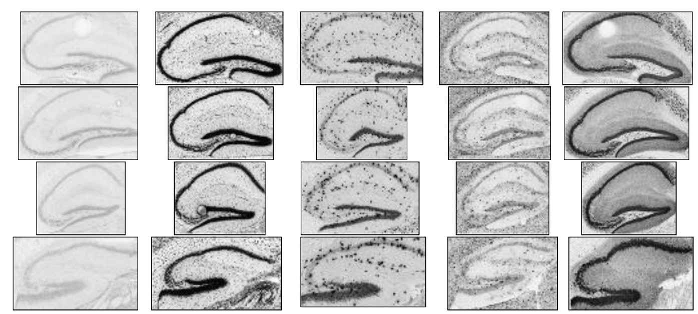
Hippocampus Localization



GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

Hippocampus Localization

- Hippocampus variability



Best Reference Slice Selection

GECCO 2014
Vancouver, July 2014 IBIS Lab, University of Parma

Hippocampus Localization

- We want to **precisely localize the hippocampus**
  - Shape is the most invariant feature
  - Usually in similar position
- The tool should be **completely automatic**
  - No human intervention
  - Applicable to a massive number of images
- We have to deal with **“difficult” images**

Deformable Models

- Curves or surfaces defined within an image domain, that move under the influence of
  - “internal forces” - related with the curve features
  - “external forces” – related with the surrounding image
- Active Shape Models (ASM):
  - Way of adding prior knowledge to Deformable Models
  - The model considers the average position of the points, and the main modes of variation found in a training set

Modified PSO

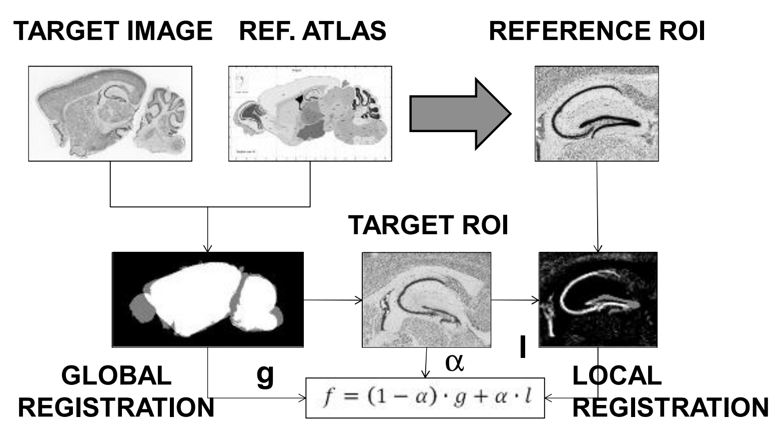
- Bio-inspired optimization algorithm developed by Kennedy and Eberhart (1995)
- Modifications:
  - Instead of using a static inertia factor *w*, its value is adapted to the fitness function of each particle [Liu et al. 2005]

$$w = \begin{cases} w_{min} + \frac{(w_{max}-w_{min})\cdot(f-f_{min})}{f_{avg}-f_{min}} & \text{if } f \leq f_{avg} \\ w_{max} & \text{if } f > f_{avg} \end{cases}$$

  - Re-initialization of a particle in case of stagnation

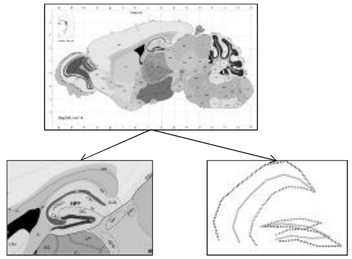
$$\begin{aligned} v_n(t) &= k \cdot randn() \\ P_n(t) &= P_n(t-1) + v_n(t) \end{aligned}$$

Best Reference Slice Selection



Localization

Localization is achieved by deforming a statistically derived model of the hippocampus to let it overlap with the corresponding structure in the brain image.



COORDINATES ROI

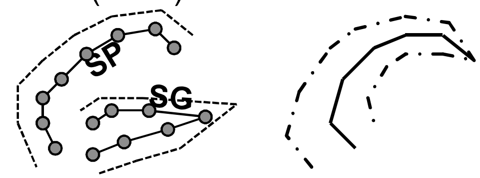
STANDARD MODEL

Localization

Two templates: SP and SG

SP has 8 points and SG 7 points.

Each template has an inner model (red) and an outer model (black)



The limits for the deformation are statistically determined (light blue dashed line).

Localization

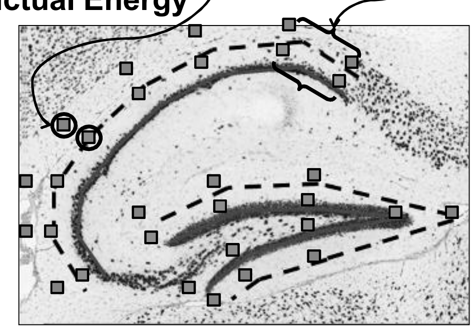
- Multimodal global continuous optimization problem
- Target function to maximize:
  - F = E – (I + C)
    - E → External Forces (minimize the intensity of pixels in the inner set and maximize the intensity of pixels in the outer set)
    - I → Internal Forces (the higher the value, the less the deformation)
    - C → Contraction Factor

Localization

- External Forces: $E = \gamma_p \cdot PE + \gamma_c \cdot CE$

Punctual Energy

Continuous Energy

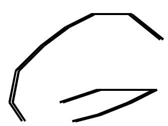


Localization

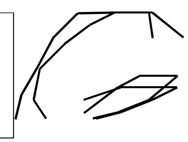
- Internal Forces:
  - aimed at keeping the model smooth (similar to the standard model) during deformation

$$I = \xi_\rho \cdot \boxed{\sqrt{\sum_{i=2}^{n} (\rho_i - \rho_{bi})^2}} + \xi_\vartheta \cdot \boxed{\sqrt{\sum_{i=2}^{n} (\vartheta_i - \vartheta_{bi})^2}}$$

Euclidean distance between the deformed model and the standard one



High values of Internal Energy

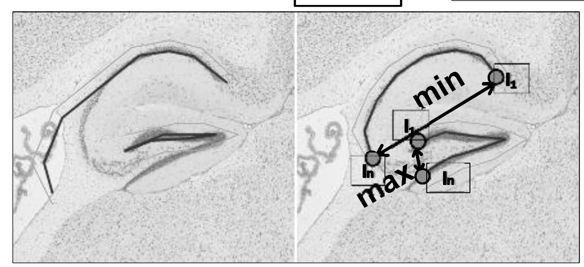


Low values of Internal Energy

Localization

Contraction Factor (avoids unfeaseable situations)

$C = \xi_c \cdot \boxed{||I_n - I_1||}$ → Points in the Inner set



$\xi c = 0$ $\xi c = 0.12$ (SP) $\xi c = -0.07$ (SG)

Experimental Results (I)

120 images | 4 methods | 50 tests/image | 250 iterations

GENETIC ALGORITHM	Pc = 0.6 | Pm = 0.09 | Pop = 80
SCATTER SEARCH	Local Search: Simulated Annealing |B|=7 |D|=8
ORIGINAL PSO	Swarm=80 particles Inertia weight = linearly decreasing*
MODIFIED PSO	$w_{min} = 0.2$ | $w_{max} = 1.0$ | $c_1 = c_2 = 2.05$ | Swarm = 80 particles

* Y. Shi and R. Eberhart. *Empirical study of Particle Swarm Optimization.* 1999.

Experimental Results (II)

Null Hypothesis (α= 0.05): there are no differences between the modified PSO and the other methods

Localization of SP

Method	Average	Std Deviation	Paired t-test
GA	108.1460	15.3365	3.05E-4
Scatter Search	87.3281	20.6248	<1.00E-16
Original PSO	110.2262	14.2808	0.2856
Modified PSO	109.6110	8.3123	-

Localization of SG

Method	Average	Std Deviation	Paired t-test
GA	140.6531	11.5352	<1.00E-16
Scatter Search	127.7413	14.5662	<1.00E-16
Original PSO	141.5991	9.3224	<1.00E-16
Modified PSO	145.2641	4.8102	-

Experimental Results (and III)

- Results
  - Perfect or good localization in 89.2% of cases (120 genes/images)
  - Noise tolerance

  - Running time for one image (non-optimized MATLAB)
    - Avg: 60.31 s
    - Std Dev: 9.01 s

Conclusions

- Metaheuristics can be used as effective tools for designing image analysis algorithms or as part of the algorithms themselves
- PSO (and Differential Evolution) are easy-implementable and efficient metaheuristics (if one does not aim at perfection..) Their 'smart search' capabilities and its intrinsic tracking ability allows it to be used in real-time applications, even more because of their intrinsic parallelism which can be "explicited" by implementing it on GPUs or multi-core processors.

BIBLIOGRAPHY

- Poli R., Cagnoni S., Valli G. (1995) ``Genetic design of optimum linear and non-linear QRS detectors'', *IEEE Transactions on Biomedical Engineering*, 42(11):1137-41.
- Cagnoni S., Dobrzeniecki A.B., Poli R., Yanch J.C. (1999) ``Genetic-algorithm-based interactive segmentation of 3D medical images'', *Image and Vision Computing Journal*, 17(12):881-896.
- Cagnoni S., Mordonini M., Sartori J. (2007) "Particle Swarm Optimization for Image Segmentation and Object Detection", Proc. *EvoWorkshops 2007*, pp. 238-247.
- L. Mussi, F. Daolio, S. Cagnoni (2011). Evaluation of Parallel Particle Swarm Optimization Algorithms within the CUDA Architecture. INFORMATION SCIENCES. 181:4642- 4657.
- Nashed, Y., Ugolotti, R., Mesejo, P., Cagnoni, S. (2012) "libCudaOptimize: an Open Source Library of GPU-based Metaheuristics", 1st GECCO Workshop on Evolutionary Computation Software Systems, Philadelphia.

BIBLIOGRAPHY

- R. Ugolotti, P. Mesejo, S. Cagnoni, M. Giacobini, F. Di Cunto (2011) Automatic hippocampus localization in histological images using PSO-based deformable models. ACM, 487- 494. In:Genetic and Evolutionary Computation Conference (GECCO 2011).
- L. Mussi, Y. S. G. Nashed, S. Cagnoni (2011) GPU-based asynchronous particle swarm optimization. ACM, 1555- 1562, ISBN:9781450305570 . In:Genetic and Evolutionary Computation Conference (GECCO 2011).
- Roberto Ugolotti, Youssef S.G. Nashed, Pablo Mesejo, Špela Ivekovič, Luca Mussi, Stefano Cagnoni (2013) "Particle Swarm Optimization and Differential Evolution for Model-based Object Detection", Applied Soft Computing Journal, Vol: 13, Pag: 3092-3105.
- P. Mesejo, R. Ugolotti, F. Di Cunto, M. Giacobini, S. Cagnoni (2013) "Automatic Hippocampus Localization in Histological Images using Differential Evolution-Based Deformable Models" Pattern Recognition Letters, Vol: 34, Pag: 299–307.